In [1]:
```
# Importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data = pd.read_csv('googleplaystore.csv')
data
```

Out[2]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Last Updated |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Photo Editor & Candy Camera & Grid & ScrapBook | ART_AND_DESIGN | 4.1 | 159 | 19M | 10,000+ | Free | 0 | Everyone | Art & Design | January 7, 2018 |
| 1 | Coloring book moana | ART_AND_DESIGN | 3.9 | 967 | 14M | 500,000+ | Free | 0 | Everyone | Art & Design;Pretend Play | January 15, 2018 |
| 2 | U Launcher Lite – FREE Live Cool Themes, Hide ... | ART_AND_DESIGN | 4.7 | 87510 | 8.7M | 5,000,000+ | Free | 0 | Everyone | Art & Design | August 1, 2018 |
| 3 | Sketch - Draw & Paint | ART_AND_DESIGN | 4.5 | 215644 | 25M | 50,000,000+ | Free | 0 | Teen | Art & Design | June 8, 2018 |
| 4 | Pixel Draw - Number Art Coloring Book | ART_AND_DESIGN | 4.3 | 967 | 2.8M | 100,000+ | Free | 0 | Everyone | Art & Design;Creativity | June 20, 2018 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 10836 | Sya9a Maroc - FR | FAMILY | 4.5 | 38 | 53M | 5,000+ | Free | 0 | Everyone | Education | July 25, 2017 |
| 10837 | Fr. Mike Schmitz Audio Teachings | FAMILY | 5.0 | 4 | 3.6M | 100+ | Free | 0 | Everyone | Education | July 6, 2018 |
| 10838 | Parkinson Exercices FR | MEDICAL | NaN | 3 | 9.5M | 1,000+ | Free | 0 | Everyone | Medical | January 20, 2017 |
| 10839 | The SCP Foundation DB fr nn5n | BOOKS_AND_REFERENCE | 4.5 | 114 | Varies with device | 1,000+ | Free | 0 | Mature 17+ | Books & Reference | January 19, 2015 |
| 10840 | iHoroscope - 2018 Daily Horoscope & Astrology | LIFESTYLE | 4.5 | 398307 | 19M | 10,000,000+ | Free | 0 | Everyone | Lifestyle | July 25, 2018 |

10841 rows × 13 columns

In [3]:
```
data.head()
```

Out[3]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Last Updated | Current Ver | An |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Photo Editor & Candy Camera & Grid & ScrapBook | ART_AND_DESIGN | 4.1 | 159 | 19M | 10,000+ | Free | 0 | Everyone | Art & Design | January 7, 2018 | 1.0.0 | a |
| 1 | Coloring book moana | ART_AND_DESIGN | 3.9 | 967 | 14M | 500,000+ | Free | 0 | Everyone | Art & Design;Pretend Play | January 15, 2018 | 2.0.0 | a |
| 2 | U Launcher Lite – FREE Live Cool Themes, Hide ... | ART_AND_DESIGN | 4.7 | 87510 | 8.7M | 5,000,000+ | Free | 0 | Everyone | Art & Design | August 1, 2018 | 1.2.4 | a |
| 3 | Sketch - Draw & Paint | ART_AND_DESIGN | 4.5 | 215644 | 25M | 50,000,000+ | Free | 0 | Teen | Art & Design | June 8, 2018 | Varies with device | 4. |
| 4 | Pixel Draw - Number Art Coloring Book | ART_AND_DESIGN | 4.3 | 967 | 2.8M | 100,000+ | Free | 0 | Everyone | Art & Design;Creativity | June 20, 2018 | 1.1 | 4. |

In [4]:
```
data.shape
```

Out[4]:
```
(10841, 13)
```

In [5]:
```
data.dtypes
```

Out[5]:
```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
App               10841 non-null object
Category          10841 non-null object
Rating            9367 non-null float64
Reviews           10841 non-null object
Size              10841 non-null object
Installs          10841 non-null object
Type              10840 non-null object
Price             10841 non-null object
Content Rating    10840 non-null object
Genres            10841 non-null object
Last Updated      10841 non-null object
Current Ver       10833 non-null object
Android Ver       10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

In [7]:
```
data.describe()
```

Out[7]:

| | Rating |
|---|---|
| count | 9367.000000 |
| mean | 4.193338 |
| std | 0.537431 |
| min | 1.000000 |
| 25% | 4.000000 |
| 50% | 4.300000 |
| 75% | 4.500000 |
| max | 19.000000 |

In [8]:
```
# Removing this row from the data because this is causing some problem 10472

data.drop(10472, axis=0, inplace=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10840 entries, 0 to 10840
Data columns (total 13 columns):
App                10840 non-null object
Category           10840 non-null object
Rating             9366 non-null float64
Reviews            10840 non-null object
Size               10840 non-null object
Installs           10840 non-null object
Type               10839 non-null object
Price              10840 non-null object
Content Rating     10840 non-null object
Genres             10840 non-null object
Last Updated       10840 non-null object
Current Ver        10832 non-null object
Android Ver        10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.2+ MB
```

In [10]:
```
data['Reviews'] = data['Reviews'].astype('int')
```

In [11]:
```
data.describe()
```

Out[11]:

| | Rating | Reviews |
|---|---|---|
| count | 9366.000000 | 1.084000e+04 |
| mean | 4.191757 | 4.441529e+05 |
| std | 0.515219 | 2.927761e+06 |
| min | 1.000000 | 0.000000e+00 |
| 25% | 4.000000 | 3.800000e+01 |
| 50% | 4.300000 | 2.094000e+03 |
| 75% | 4.500000 | 5.477550e+04 |
| max | 5.000000 | 7.815831e+07 |

In [12]:
```
# Taking size column and make it numeric

data['Size'].value_counts()
```

```
Varies with device    1695
11M                    198
12M                    196
14M                    194
13M                    191
                      ...
921k                     1
942k                     1
314k                     1
705k                     1
39k                      1
Name: Size, Length: 461, dtype: int64
```

In [13]:
```
data['Size'].isnull().sum()
```

Out[13]:
```
0
```

In [14]:
```
#Checking the number of values in three different categories in Size

print("Number of M in Size Column",
data['Size'].loc[data['Size'].str.contains('M')].value_counts().sum())
print("Number of k in Size Column",
data['Size'].loc[data['Size'].str.contains('k')].value_counts().sum())
print("Number of Varies with device in Size Column",
data['Size'].loc[data['Size'].str.contains('Varies with device')].value_counts().sum())
```

```
Number of M in Size Column 8829
Number of k in Size Column 316
Number of Varies with device in Size Column 1695
```

In [16]:
```
#Convert the whole size of the column into bytes

### Defining a Function

def convert_into_bytes(column_name):
    if isinstance(column_name, str):
        if 'k' in column_name:
            return float(column_name.replace("k", "")) * 1024
        elif 'M' in column_name:
            return float(column_name.replace("M", "")) * 1024 * 1024
        elif 'Varies with device' in column_name:
            return np.nan
    return column_name
```

In [18]:
```
data['Size'] = data['Size'].apply(convert_into_bytes)
data['Size']
```

Out[18]:
```
0        19922944.0
1        14680064.0
2         9122611.2
3        26214400.0
4         2936012.8
             ...
10836    55574528.0
10837     3774873.6
10838     9961472.0
10839           NaN
10840    19922944.0
Name: Size, Length: 10840, dtype: float64
```

In [84]:
```
# Remove + sign
# Remove , from the values
#Convert the column in to integers

## Define a function to deal with installs column

def installs(install):
    if isinstance(install, str):
        if '+' in install:
            return install.replace("+", "")
    return int(install)
```

In [85]:
```
data['Installs'] = data['Installs'].apply(installs)
```

In [86]:
```
data['Installs'] = data['Installs'].apply(lambda x: x.replace(',', '') if ',' in str(x)
else x)
```

In [87]:
```
data['Installs'] = data['Installs'].astype('int')
```

In [88]:
```
data['Installs'].value_counts()
```

Out[88]:
```
1000000       1487
10000000      1132
100000        1129
10000         1031
1000           888
100            709
5000000        683
500000         516
50000          473
5000           468
10             384
100000000      369
500            328
50000000       272
50             204
5               82
1               67
500000000       61
1000000000      49
0               14
Name: Installs, dtype: int64
```

In [89]:
```
# making a new column called 'Installs_category' which will have the category of the installs

bins = [-1, 0, 10, 1000, 10000, 100000, 1000000, 10000000, 10000000000]
labels=['no', 'Very low', 'Low', 'Moderate', 'More than moderate', 'High', 'Very High', 'Top Notch']
data['Installs_category'] = pd.cut(data['Installs'], bins=bins, labels=labels)
```

In [90]:
```
data['Installs_category'].value_counts()
```

Out[90]:
```
Low                  2129
High                 2003
Very High            1815
More than moderate   1602
Moderate             1499
Top Notch             751
Very low              533
no                     14
Name: Installs_category, dtype: int64
```

In [91]:
```
data.head(4)
```

Out[91]:

| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Last Updated | Current Ver |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Photo Editor & Candy Camera & Grid & ScrapBook | ART_AND_DESIGN | 4.1 | 159 | 19922944.0 | 10000 | Free | 0.0 | Everyone | Art & Design | January 7, 2018 | 1.0.0 |
| 1 | Coloring book moana | ART_AND_DESIGN | 3.9 | 967 | 14680064.0 | 500000 | Free | 0.0 | Everyone | Art & Design;Pretend Play | January 15, 2018 | 2.0.0 |
| 2 | U Launcher Lite – FREE Live Cool Themes, Hide ... | ART_AND_DESIGN | 4.7 | 87510 | 9122611.2 | 5000000 | Free | 0.0 | Everyone | Art & Design | August 1, 2018 | 1.2.4 |
| 3 | Sketch - Draw & Paint | ART_AND_DESIGN | 4.5 | 215644 | 26214400.0 | 50000000 | Free | 0.0 | Teen | Art & Design | June 8, 2018 | Varies with device |

In [28]:
```
# Taking Price column

data['Price'].unique()
```

Out[28]:
```
array(['0', '$4.99', '$3.99', '$6.99', '$1.49', '$2.99', '$7.99', '$5.99',
       '$3.49', '$1.99', '$9.99', '$7.49', '$0.99', '$9.00', '$5.49',
       '$10.00', '$24.99', '$11.99', '$79.99', '$16.99', '$14.99',
       '$1.00', '$29.99', '$12.99', '$2.49', '$10.99', '$1.50', '$19.99',
       '$15.99', '$33.99', '$74.99', '$39.99', '$3.95', '$4.49', '$1.70',
       '$8.99', '$2.00', '$3.88', '$25.99', '$399.99', '$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61', '$2.50',
       '$1.59', '$6.49', '$1.29', '$5.00', '$13.99', '$299.99', '$379.99',
       '$37.99', '$18.99', '$389.99', '$19.90', '$8.49', '$1.75',
       '$14.00', '$4.85', '$46.99', '$109.99', '$154.99', '$3.08',
       '$2.59', '$4.80', '$1.96', '$19.40', '$3.90', '$4.59', '$15.46',
       '$3.04', '$4.29', '$2.60', '$3.28', '$4.60', '$28.99', '$2.95',
       '$2.90', '$1.97', '$200.00', '$89.99', '$2.56', '$30.99', '$3.61',
       '$394.99', '$1.26', '$1.20', '$1.04'], dtype=object)
```

In [29]:
```
def adjust_price(price):
    if isinstance(price, str):
        if '$' in price:
            return price.replace("$", "")
    return price
```

In [30]:
```
data['Price'] = data['Price'].apply(adjust_price)
```

In [31]:
```
data['Price'].unique()
```

Out[31]:
```
array(['0', '4.99', '3.99', '6.99', '1.49', '2.99', '7.99', '5.99',
       '3.49', '1.99', '9.99', '7.49', '0.99', '9.00', '5.49', '10.00',
       '24.99', '11.99', '79.99', '16.99', '14.99', '1.00', '29.99',
       '12.99', '2.49', '10.99', '1.50', '19.99', '15.99', '33.99',
       '74.99', '39.99', '3.95', '4.49', '1.70', '8.99', '2.00', '3.88',
       '25.99', '399.99', '17.99', '400.00', '3.02', '1.76', '4.84',
       '4.77', '1.61', '2.50', '1.59', '6.49', '1.29', '5.00', '13.99',
       '299.99', '379.99', '37.99', '18.99', '389.99', '19.90', '8.49',
       '1.75', '14.00', '4.85', '46.99', '109.99', '154.99', '3.08',
       '2.59', '4.80', '1.96', '19.40', '3.90', '4.59', '15.46', '3.04',
       '4.29', '2.60', '3.28', '4.60', '28.99', '2.95', '2.90', '1.97',
       '200.00', '89.99', '2.56', '30.99', '3.61', '394.99', '1.26',
       '1.20', '1.04'], dtype=object)
```

In [92]:
```
data['Price'].dtype
```

Out[92]:
```
dtype('float64')
```

In [93]:
```
data['Price'] = data['Price'].astype('float')
```

In [94]:
```
data.describe()
```

Out[94]:

| | Rating | Reviews | Size | Installs | Price |
|---|---|---|---|---|---|
| count | 8886.000000 | 1.034600e+04 | 8.821000e+03 | 1.034600e+04 | 10346.000000 |
| mean | 4.187959 | 4.063338e+05 | 2.234121e+07 | 1.417266e+07 | 1.031561 |
| std | 0.522428 | 2.698179e+06 | 2.364101e+07 | 8.028090e+07 | 16.287252 |
| min | 1.000000 | 0.000000e+00 | 8.704000e+03 | 0.000000e+00 | 0.000000 |
| 25% | 4.000000 | 3.200000e+01 | 4.928307e+06 | 1.000000e+03 | 0.000000 |
| 50% | 4.300000 | 1.688500e+03 | 1.363149e+07 | 1.000000e+05 | 0.000000 |
| 75% | 4.500000 | 4.659825e+04 | 3.040870e+07 | 1.000000e+06 | 0.000000 |
| max | 5.000000 | 7.815831e+07 | 1.048576e+08 | 1.000000e+09 | 400.000000 |

In [36]:
```
data.head()
```

Out[36]:

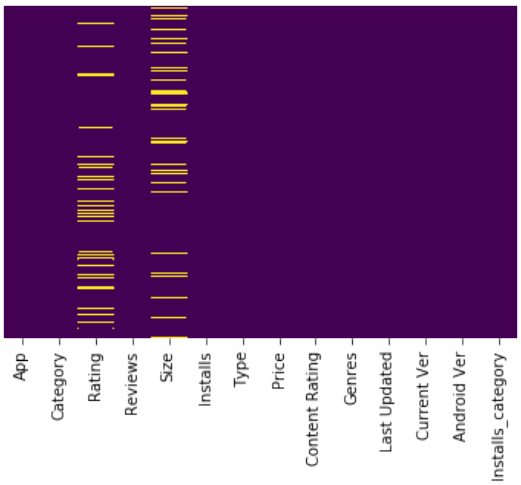
| | App | Category | Rating | Reviews | Size | Installs | Type | Price | Content Rating | Genres | Last Updated | Current Ver |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Photo Editor & Candy Camera & Grid & ScrapBook | ART_AND_DESIGN | 4.1 | 159 | 19922944.0 | 10000 | Free | 0.0 | Everyone | Art & Design | January 7, 2018 | 1.0.0 |
| 1 | Coloring book moana | ART_AND_DESIGN | 3.9 | 967 | 14680064.0 | 500000 | Free | 0.0 | Everyone | Art & Design;Pretend Play | January 15, 2018 | 2.0.0 |
| 2 | U Launcher Lite – FREE Live Cool Themes, Hide ... | ART_AND_DESIGN | 4.7 | 87510 | 9122611.2 | 5000000 | Free | 0.0 | Everyone | Art & Design | August 1, 2018 | 1.2.4 |
| 3 | Sketch - Draw & Paint | ART_AND_DESIGN | 4.5 | 215644 | 26214400.0 | 50000000 | Free | 0.0 | Teen | Art & Design | June 8, 2018 | Varies with device |
| 4 | Pixel Draw - Number Art Coloring Book | ART_AND_DESIGN | 4.3 | 967 | 2936012.8 | 100000 | Free | 0.0 | Everyone | Art & Design;Creativity | June 20, 2018 | 1.1 |

In [95]:
```
# Missing Values

data.isnull().sum().sort_values(ascending=False)
```

Out[95]:
```
Size                1525
Rating              1460
Installs_category      0
Android Ver            0
Current Ver            0
Last Updated           0
Genres                 0
Content Rating         0
Price                  0
Type                   0
Installs               0
Reviews                0
Category               0
App                    0
dtype: int64
```

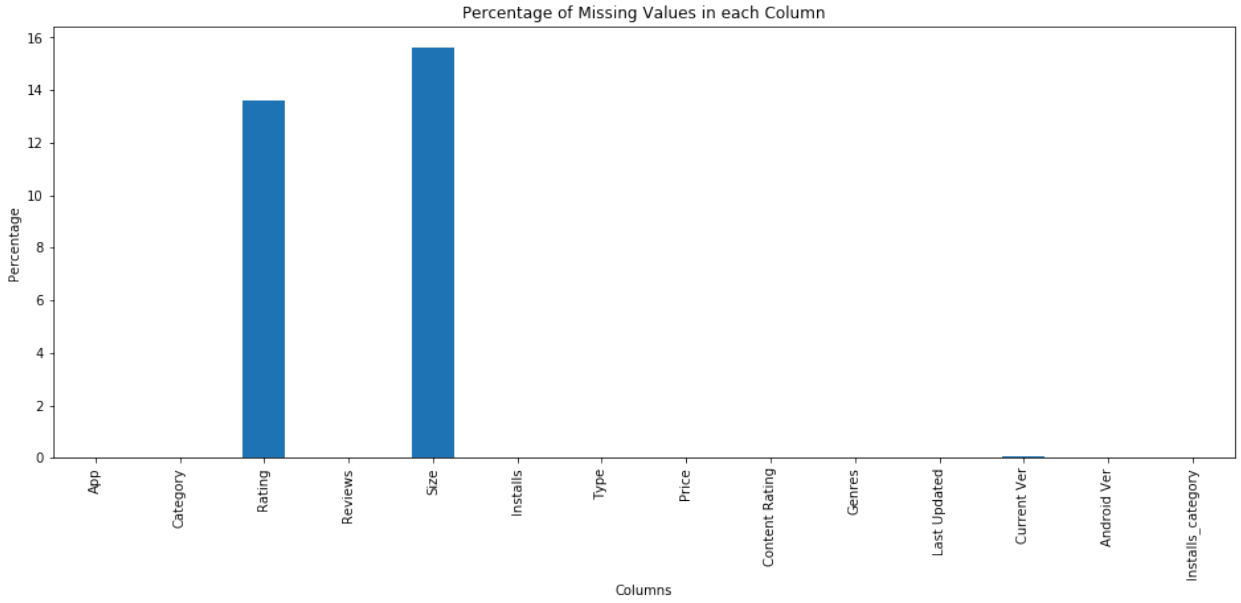
In [96]:
```
### Plot Missing Values

sns.heatmap(data.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```

Out[96]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ac61b235c8>
```



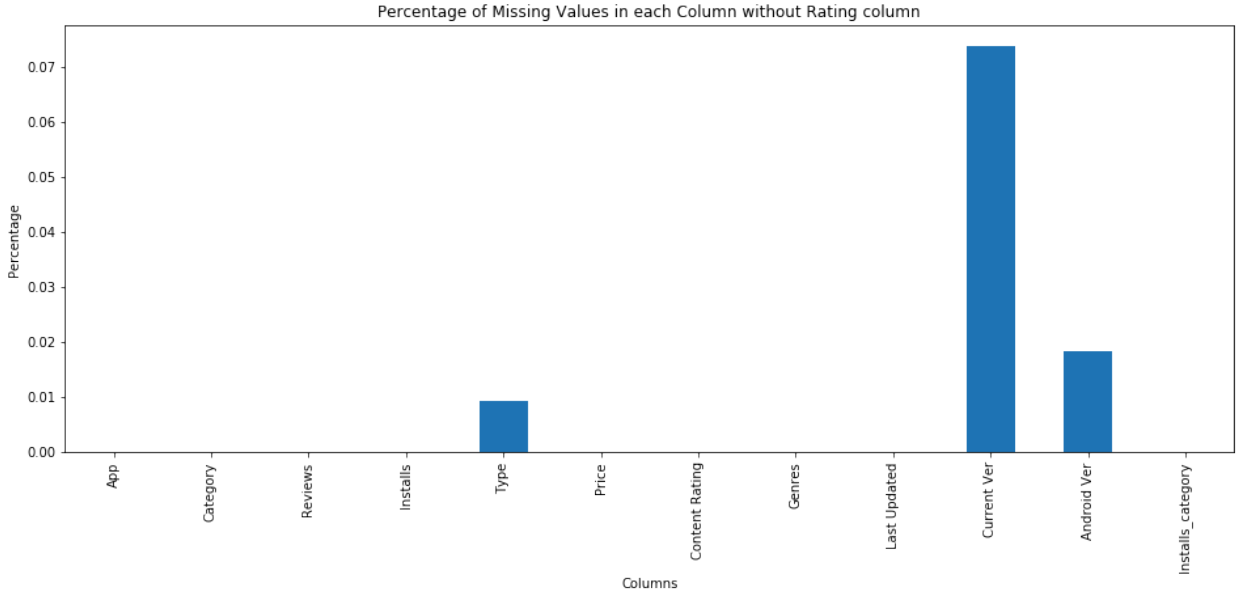
In [40]:
```
# make figure size

plt.figure(figsize=(16, 6))

# plot the null values by their percentage in each column

missing_percentage = data.isnull().sum()/len(data)*100

missing_percentage.plot(kind='bar')

# add the labels

plt.xlabel('Columns')
plt.ylabel('Percentage')
plt.title('Percentage of Missing Values in each Column')
```

Out[40]:
```
Text(0.5, 1.0, 'Percentage of Missing Values in each Column')
```




In [41]:
```
plt.figure(figsize=(16, 6)) # make figure size

missing_percentage[missing_percentage < 1].plot(kind='bar') # plot the null values by their percentage in

plt.xlabel('Columns') # add the x-axis labels
plt.ylabel('Percentage') # add the labels for y-axis
plt.title('Percentage of Missing Values in each Column without Rating column') # add the title for the plo
```

Out[41]:
```
Text(0.5, 1.0, 'Percentage of Missing Values in each Column without Rating column')
```



In [119]:
```
##Dealing with the missing values
##We can not impute the Rating column as is is directly linked with the installationcolumn.
#To test this Hypothesis we need to plot the Rating column with the
#Installs and Size columns and statistically test it using pearson correlation test.

data.columns
```

Out[119]:
```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver', 'Installs_category'],
      dtype='object')
```

In [120]:
```
numeric_cols = [i for i in data.columns if data[i].dtype != 'object' ] # make a list of numeric columns
```

In [121]:
```
numeric_cols.remove("Installs_category")
```

In [122]:
```
numeric_cols
```

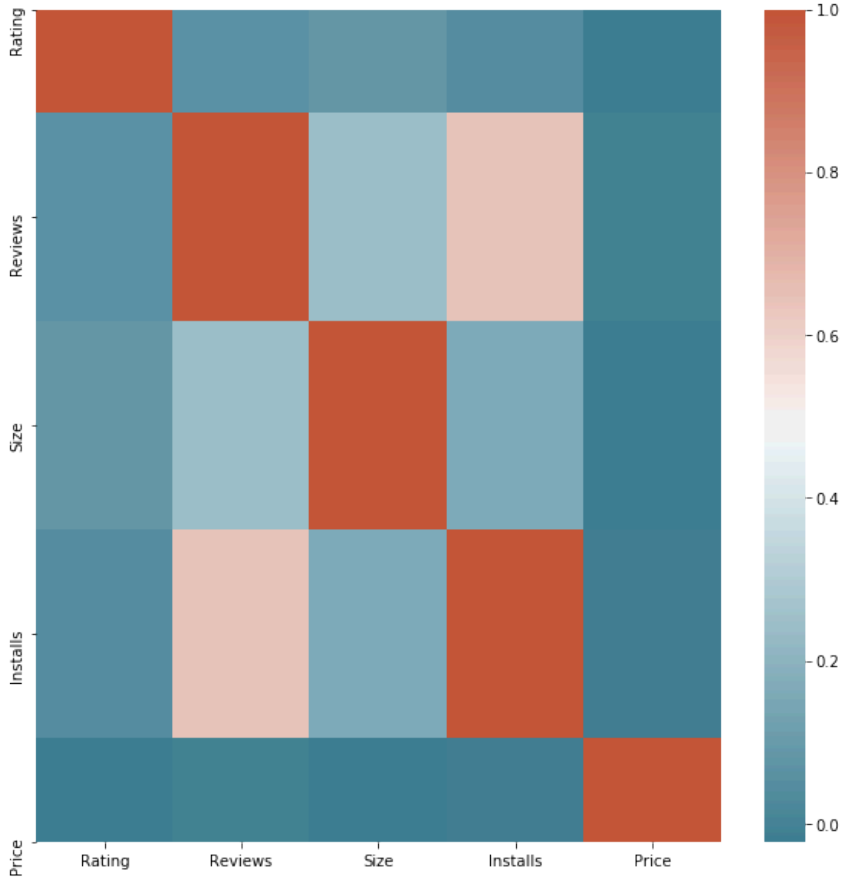
Out[122]:
```
['Rating', 'Reviews', 'Size', 'Installs', 'Price']
```

In [123]:
```
corr = data[numeric_cols].corr()
corr
```

Out[123]:

| | Rating | Reviews | Size | Installs | Price |
|---|---|---|---|---|---|
| Rating | 1.000000 | 0.068724 | 0.081874 | 0.050869 | -0.022371 |
| Reviews | 0.068724 | 1.000000 | 0.237853 | 0.634987 | -0.009424 |
| Size | 0.081874 | 0.237853 | 1.000000 | 0.168805 | -0.023820 |
| Installs | 0.050869 | 0.634987 | 0.168805 | 1.000000 | -0.011155 |
| Price | -0.022371 | -0.009424 | -0.023820 | -0.011155 | 1.000000 |

In [48]:
```
plt.figure(figsize=(10, 10))
sns.heatmap(corr, cmap=sns.diverging_palette(220, 20, as_cmap=True))
plt.show()
```



In [97]:
```
# remove rows containing NaN or infinite values (Important to calculate Pearson's R)

data_clean = data.dropna()
```

In [98]:
```
from scipy import stats
```

In [99]:
```
# calculate Pearson's R between Rating and Installs

pearson_r, _ = stats.pearsonr(data_clean['Reviews'], data_clean['Installs'])
print(f"Pearson's R between Reviews and Installs: {pearson_r:.4f}")
```

```
Pearson's R between Reviews and Installs: 0.6320
```

In [100]:
```
# remove the rows having null values in the 'Current Ver', 'Android Ver', 'Category','Type' and 'Genres' c

data.dropna(subset=['Current Ver', 'Android Ver', 'Category', 'Type', 'Genres'],
inplace=True)
```

In [101]:
```
# length after removing null values

print(f"Length of the dataframe after removing null values: {len(data)}")
```

```
Length of the dataframe after removing null values: 10346
```

In [56]:
```
# use groupby function to find the trend of Rating in each Installs_category

data.groupby('Installs_category')['Rating'].describe()
```

Out[56]:

| Installs_category | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| no | 0.0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| Very low | 81.0 | 4.637037 | 0.845199 | 1.0 | 4.8 | 5.0 | 5.0 | 5.0 |
| Low | 1278.0 | 4.170970 | 0.825605 | 1.0 | 3.8 | 4.4 | 4.8 | 5.0 |
| Moderate | 1440.0 | 4.035417 | 0.604428 | 1.4 | 3.8 | 4.2 | 4.5 | 5.0 |
| More than moderate | 1616.0 | 4.093255 | 0.505619 | 1.6 | 3.9 | 4.2 | 4.5 | 4.9 |
| High | 2113.0 | 4.207525 | 0.376594 | 1.8 | 4.0 | 4.3 | 4.5 | 4.9 |
| Very High | 2004.0 | 4.287076 | 0.294902 | 2.0 | 4.1 | 4.3 | 4.5 | 4.9 |
| Top Notch | 828.0 | 4.374396 | 0.193726 | 3.1 | 4.3 | 4.4 | 4.5 | 4.8 |

In [102]:
```
data['Rating'].isnull().sum()
```

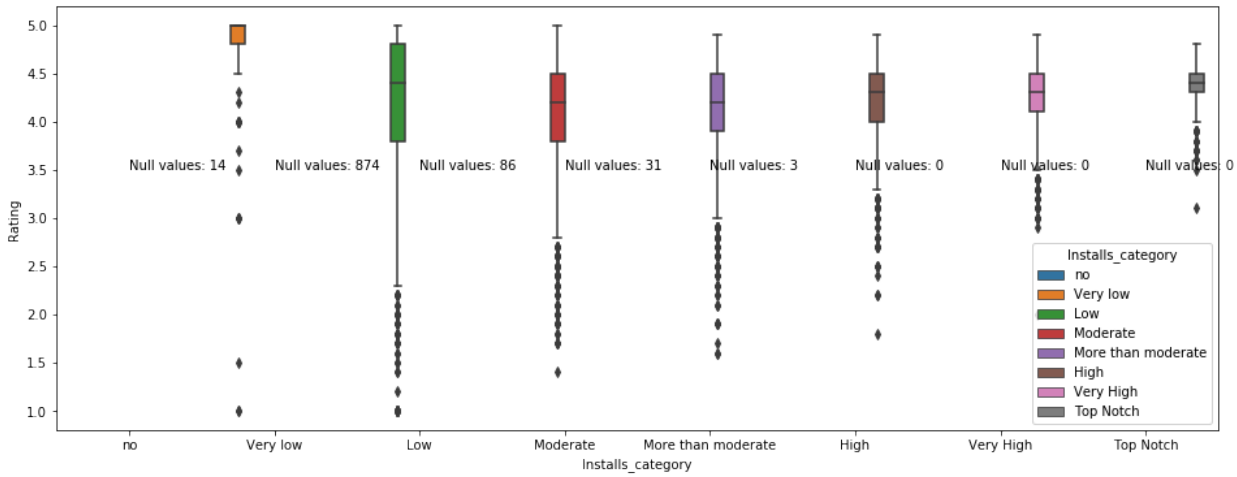
Out[102]:
```
1460
```

In [124]:
```
# in which Install_category the Rating has NaN values

data['Installs_category'].loc[data['Rating'].isnull()].value_counts()
```

Out[124]:
```
Low                  874
Very low             452
Moderate              86
More than moderate    31
no                    14
High                   3
Top Notch              0
Very High              0
Name: Installs_category, dtype: int64
```

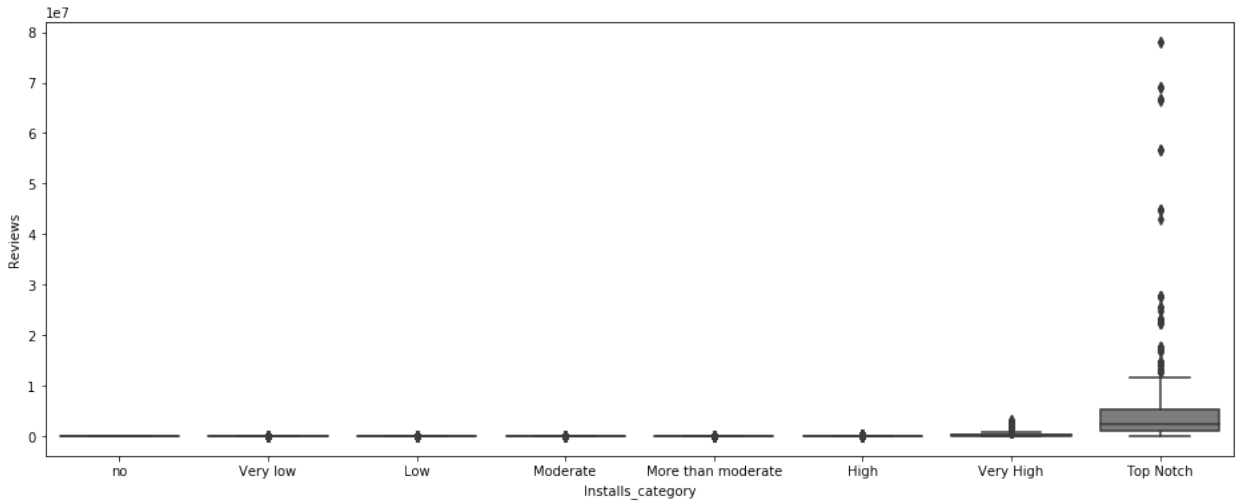
In [60]:
```
# plot the boxplot of Rating in each Installs_category

plt.figure(figsize=(16, 6)) # make figure size
sns.boxplot(x='Installs_category', y='Rating', hue='Installs_category', data=data) #plot the boxplot

# add the text of number of null values in each category

plt.text(0, 3.5, 'Null values: 14')
plt.text(1, 3.5, 'Null values: 874')
plt.text(2, 3.5, 'Null values: 86')
plt.text(3, 3.5, 'Null values: 31')
plt.text(4, 3.5, 'Null values: 3')
plt.text(5, 3.5, 'Null values: 0')
plt.text(6, 3.5, 'Null values: 0')
plt.text(7, 3.5, 'Null values: 0')
```

Out[60]:
```
Text(7, 3.5, 'Null values: 0')
```



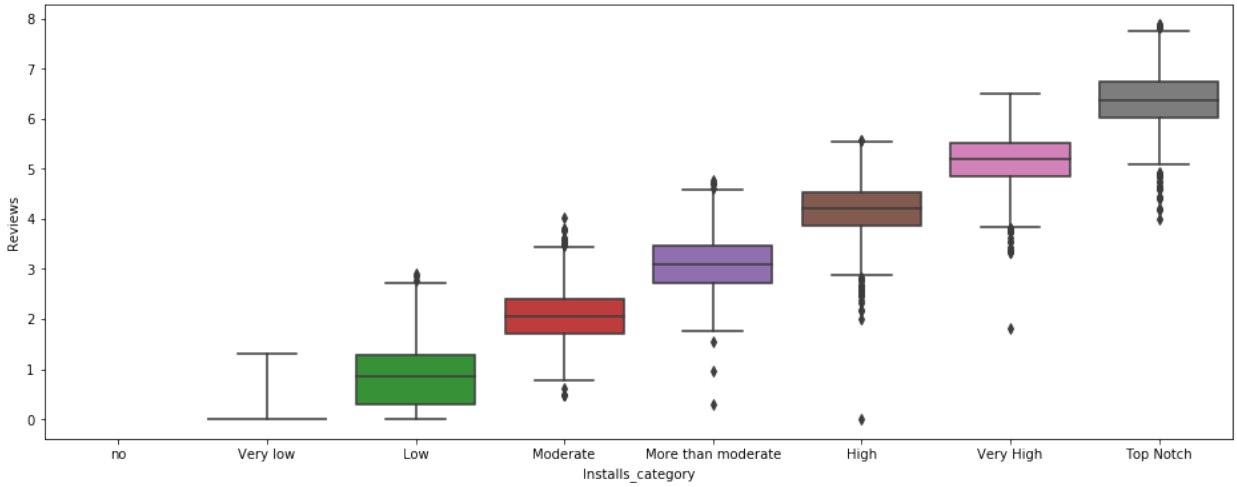

In [61]:
```
# let's plot the same plots for Reviews column as well

plt.figure(figsize=(16, 6)) # make figure size
sns.boxplot(x='Installs_category', y= 'Reviews', data=data) # plot the boxplot
```

Out[61]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ac5e642988>
```



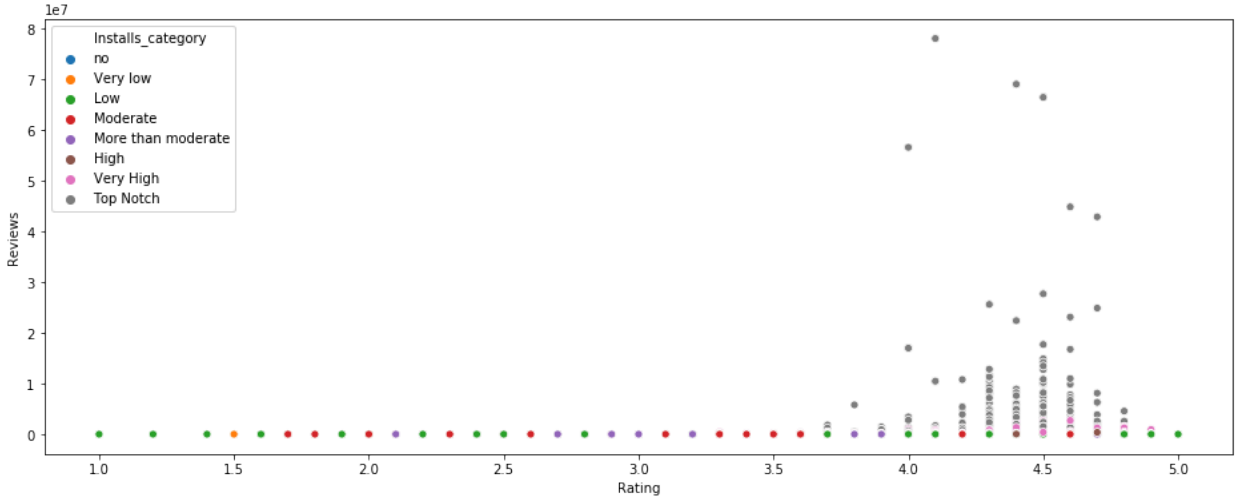
In [63]:
```
# let's plot the same plots for Reviews column as well

plt.figure(figsize=(16, 6)) # make figure size
sns.boxplot(x='Installs_category', y= np.log10(data['Reviews']), data=data) # plot the boxplot
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ac5f265d88>
```



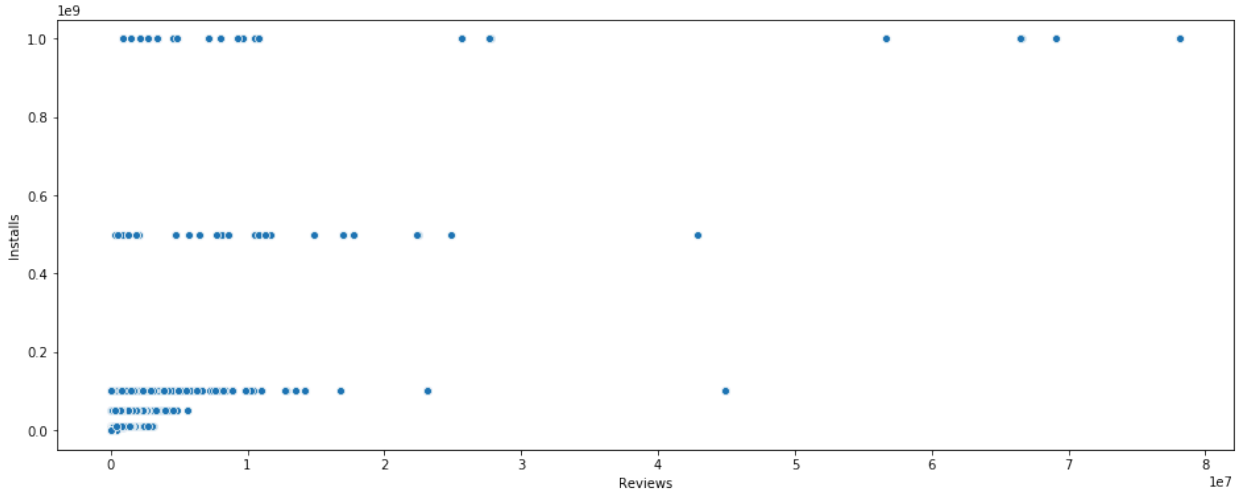

In [65]:
```
# Draw a scatter plot between Rating, Reviews and Installs

plt.figure(figsize=(16, 6)) # make figure size
sns.scatterplot(x='Rating', y='Reviews', hue='Installs_category', data=data) # plot the scatter plot
```

Out[65]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ac5ec0d208>
```



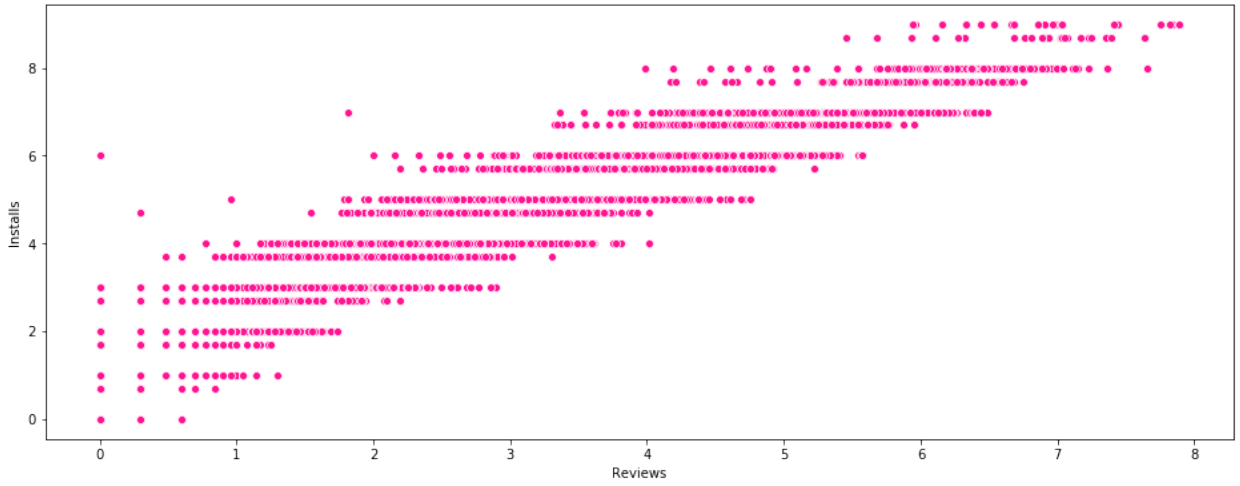
In [66]:
```
# plot reviews and installs in a scatter plot

plt.figure(figsize=(16, 6)) # make figure size
sns.scatterplot(x='Reviews', y='Installs', data=data) # plot the scatter plot
```

Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1ac5ecafa88>



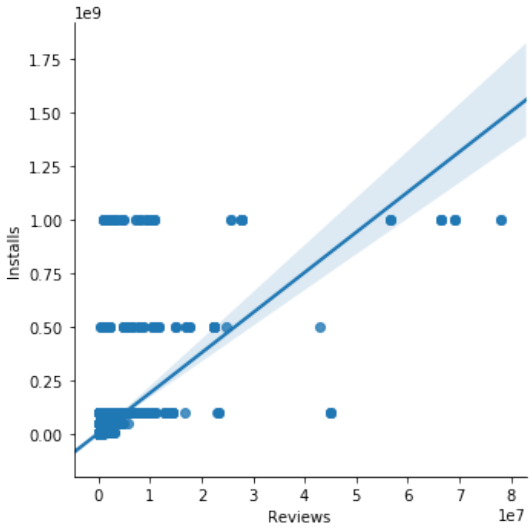

In [70]:
```
# plot reviews and installs in a scatter plot

plt.figure(figsize=(16, 6)) # make figure size
sns.scatterplot(x=np.log10(data['Reviews']), y=np.log10(data['Installs']), data=data, color = 'deeppink')
```

Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x1ac600500c8>



In [72]:
```
# plot reviews and installs in a scatter plot with trend line

plt.figure(figsize=(16, 6)) # make figure size
sns.lmplot(x='Reviews', y='Installs', data=data) # plot the scatter plot with trend line
```

Out[72]:
```
<seaborn.axisgrid.FacetGrid at 0x1ac603f4108>

<Figure size 1152x432 with 0 Axes>
```




In [73]:
```
# find duplicate if any

data.duplicated().sum()
```

Out[73]:
```
483
```

In [104]:
```
# let's check for number of duplicates

for col in data.columns:
    print(f"Number of duplicates in {col} column are: {data[col].duplicated().sum()}")
```

```
Number of duplicates in App column are: 698
Number of duplicates in Category column are: 10313
Number of duplicates in Rating column are: 10306
Number of duplicates in Reviews column are: 4347
Number of duplicates in Size column are: 9890
Number of duplicates in Installs column are: 10326
Number of duplicates in Type column are: 10344
Number of duplicates in Price column are: 10254
Number of duplicates in Content Rating column are: 10340
Number of duplicates in Genres column are: 10227
Number of duplicates in Last Updated column are: 8970
Number of duplicates in Current Ver column are: 7515
Number of duplicates in Android Ver column are: 10313
Number of duplicates in Installs_category column are: 10338
```

In [125]:
```
# remove the duplicates

#data.drop_duplicates(inplace=True)
```

In [130]:
```
# category with highest number of Prices

data.groupby('Category')['Installs'].sum().sort_values(ascending=False).head(10)
```

Out[130]:
```
Category
GAME                 31544024415
COMMUNICATION        24152276251
SOCIAL               12513867902
PRODUCTIVITY         12463091369
TOOLS                11452271905
FAMILY               10041632405
PHOTOGRAPHY           9721247655
TRAVEL_AND_LOCAL      6361887146
VIDEO_PLAYERS         6222002720
NEWS_AND_MAGAZINES    5393217760
Name: Installs, dtype: int64
```

In [131]:
```
# Category with highest average Rating

data.groupby('Category')['Rating'].mean().sort_values(ascending=False).head(10)
```

Out[131]:
```
Category
EVENTS               4.435556
ART_AND_DESIGN       4.377049
EDUCATION            4.375969
BOOKS_AND_REFERENCE  4.347458
PERSONALIZATION      4.333117
PARENTING            4.300000
GAME                 4.281285
BEAUTY               4.278571
HEALTH_AND_FITNESS   4.261450
SOCIAL               4.254918
Name: Rating, dtype: float64
```

In [ ]: